Community Detection: Overlapping Communities

CS224W: Social and Information Network Analysis
Jure Leskovec, Stanford University
http://cs224w.stanford.edu
Overlapping Communities

- Non-overlapping vs. overlapping communities
A node belongs to many social circles
Two nodes belong to the same community if they can be connected through adjacent $k$-cliques:

- **$k$-clique:**
  - Fully connected graph on $k$ nodes

- **Adjacent $k$-cliques:**
  - Overlap in $k-1$ nodes

- **$k$-clique community**
  - Set of nodes that can be reached through a sequence of adjacent $k$-cliques

[11/10/2010]

Two nodes belong to the same community if they can be connected through adjacent $k$-cliques:
CPM: Steps

- **Clique Percolation Method:**
  - Find maximal-cliques (not \(k\)-cliques!)
  - **Clique overlap graph:**
    - Each clique is a node
    - Connect two cliques if they overlap in at least \(k-1\) nodes
  - **Communities:**
    - Connected components of the clique overlap matrix
- **How to set \(k\)?**
  - Set \(k\) so that we get the “richest” (most widely distributed cluster sizes) community structure
CPM method: Example

- Start with graph
- Find maximal cliques
- Create clique overlap matrix
- Threshold the matrix at value $k-1$
  - If $a_{ij} < k-1$ set 0
- Communities are the connected components of the thresholded matrix
Example: Phone-Call Network

Communities in a “tiny” part of a phone call network of 4 million users

[Palla et al., ’07]
Example: Website

[Farkas et. al. 07]
How to Find Maximal Cliques?

- **No nice way, NP-hard combinatorial problem**
- **Maximal clique:** clique that can’t be extended
  - \{a,b,c\} is a clique but not maximal clique
  - \{a,b,c,d\} is maximal clique
- **Algorithm:** Sketch
  - Start with a seed node
  - Expand the clique around the seed
  - Once the clique cannot be further expanded we found the maximal clique
- **Note:**
  - This will generate the same clique multiple times
How to Find Maximal Cliques?

- **Start with a seed vertex “a”**
- **Goal:** Find the maximal clique $Q$ “a” belongs to
  - **Observation:**
    - If some “x” belongs to $Q$ then it is a member of “a”
      - **Why?** If $a, x \in Q$ but not $a-x$, then $Q$ is not a clique!
  - **Recursive algorithm:**
    - $Q$ … current clique
    - $R$ … candidate vertices to expand the clique to
  - **Example:** Start with “a” and expand around it

\[
Q = \{a\} \quad \{a, b\} \quad \{a, b, c\} \quad \text{bktrack} \quad \{a, b, d\} \\
R = \{b, c, d\} \quad \{b, c, d\} \quad \{d\} \cap \Gamma(c) = \{} \quad \{c\} \cap \Gamma(d) = \{} \\
\cap \Gamma(b) = \{c, d\}
\]

Steps of the recursive algorithm

\[
\Gamma(u) \ldots \text{neighbor set of } u
\]
How to Find Maximal Cliques?

- **Q** ... current clique
- **R** ... candidate vertices

**Expand (R, Q)**

- **while** **R** ≠ **{ }**
  - **p** = vertex in **R**
  - **Q_p** = **Q** ∪ **{p}**
  - **R_p** = **R** ∩ $$\Gamma(p)$$
  - **if** **R_p** ≠ **{ }**: Expand (**R_p**, **Q_p**)
  - else: output **Q_p**
  - **R** = **R** − **{p}**

---

Start: Expand(V, {})

- **R**={a,…f}, **Q**={}
- **p** = {a}
- **Q_p** = {a}
- **R_p** = {b,d}

Expand(R_p, Q):  
  - **R** = {b,d}, **Q**={a}
  - **p** = {b}
  - **Q_p** = {a,b}
  - **R_p** = {d}

Expand(R_p, Q):  
  - **R** = {d}, **Q**={a,b}
  - **p** = {d}
  - **Q_p** = {a,b,d}
  - **R_p** = {}: output {a,b,d}

p = {d}
- **Q_p** = {a,d}
- **R_p** = {b}

Expand(R_p, Q):  
  - **R** = {b}, **Q**={a,d}
  - **p** = {b}
  - **Q_p** = {a,d}
  - **R_p** = {}: output {a,d,b}
How to Find Maximal Cliques?

- **Q** ... current clique
- **R** ... candidate vertices

**Expand (R, Q)**

- **while** \( R \neq \{\} \)
  - \( p = \text{vertex in } R \)
  - \( Q_p = Q \cup \{p\} \)
  - \( R_p = R \cap \Gamma(p) \)
  - **if** \( R_p \neq \{\} \): Expand \((R_p, Q_p)\)
  - **else**: output \( Q_p \)
  - \( R = R - \{p\} \)

Start: Expand(V, \{\})

\[ \begin{align*}
R &= \{a, \ldots f\}, \; Q = \{\} \\
p &= \{b\} \\
Q_p &= \{b\} \\
R_p &= \{a, c, d\} \\
\text{Expand}(R_p, Q): \\
R &= \{a, c, d\}, \; Q = \{b\} \\
p &= \{a\} \\
Q_p &= \{b, a\} \\
R_p &= \{d\} \\
\end{align*} \]

\[ \begin{align*}
\text{Expand}(R_p, Q): \\
R &= \{d\}, \; Q = \{b, a\} \\
p &= \{d\} \\
Q_p &= \{b, a, d\} \\
R_p &= \{\} : \text{output } \{b, a, d\} \\
p &= \{c\} \\
Q_p &= \{b, c\} \\
R_p &= \{d\} \\
\text{Expand}(R_p, Q): \\
R &= \{d\}, \; Q = \{b, c\} \\
p &= \{d\} \\
Q_p &= \{b, c, d\} \\
R_p &= \{\} : \text{output } \{b, c, d\} \\
\end{align*} \]
How to prevent maximal cliques to be generated multiple times?

- Only output cliques that are lexicographically minimum
  - \{a,b,c\} < \{b,a,c\}

- **Even better**: Only expand to the nodes higher in the lexicographical order

Start: Expand(V, \{\})
R={a,...,f}, Q=\{}
p = \{a\}
Q_p = \{a\}
R_p = \{b,d\}

Expand(R_p, Q):
R = \{b,d\}, Q=\{a\}
p = \{b\}
Q_p = \{a,b\}
R_p = \{d\}

Expand(R_p, Q):
R = \{d\}, Q=\{a,b\}
p = \{d\}
Q_p = \{a,b,d\}
R_p = \{\} : output \{a,b,d\}
p = \{d\}
Q_p = \{a,d\}
R_p = \{b\} : Don’t expand

b < d
How to Model Networks with Communities?
Let’s rethink what we are doing...

- Given a network
- Want to find communities!

Need to:

- Formalize the notion of a community
- Need an algorithm that will find sets of nodes that are “good” communities

More generally:

- How to think about clusters in large networks?
What is a good cluster?
- Many edges internally
- Few pointing outside

Formally, conductance:

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\{A(S), A(\overline{S})\}}$$

Where: $A(S)$....volume

$A(S) = \sum_{i \in S} \sum_{j \in V} A_{ij}$

Small $\Phi(S)$ corresponds to good clusters
How community like is a set of nodes?
A good cluster $S$ has
- Many edges internally
- Few edges pointing outside

Simplest objective function:
**Conductance**

$$\phi(S) = \frac{\left| \{(i, j) \in E; i \in S, j \notin S \} \right|}{\sum_{s \in S} d_s}$$

Small **conductance** corresponds to good clusters
Define:

Network community profile (NCP) plot

Plot the score of best community of size $k$

$$
\Phi(k) = \min_{S \subseteq V, |S| = k} \phi(S)
$$

![Diagram showing log $\Phi(k)$ vs. Community size, log $k$]
How to (Really) Compute NCP?

- Run the favorite clustering method
- Each dot represents a cluster
- For each size find “best” cluster
NCP Plot: Meshes

- Meshes, grids, dense random graphs:

![Graph showing conductance vs. number of nodes in the cluster for different structures: Random graph, Cube, Grid, Chain.]

- d-dimensional meshes

- California road network
Collaborations between scientists in networks

[Newman, 2005]
Natural hypothesis about NCP:

- **NCP of real networks slopes downward**
- **Slope of the NCP corresponds to the “dimensionality” of the network**

**What about large networks?**

---

**Social nets**
- **LIVEJOURNAL**
  - Nodes: 4,843,953
  - Edges: 42,845,684
  - Description: Blog friendships [5]
- **EPINIONS**
  - Nodes: 75,877
  - Edges: 405,739
  - Description: Trust network [28]
- **CA-DBLP**
  - Nodes: 317,080
  - Edges: 1,049,866
  - Description: Co-author network [5]

**Information (citation) networks**
- **CIT-HEP-TH**
  - Nodes: 27,400
  - Edges: 352,021
  - Description: ArXiv hep-th [14]
- **AMAZONPROD**
  - Nodes: 524,371
  - Edges: 1,491,793
  - Description: Amazon products [8]

**Web graphs**
- **WEB-GOOGLE**
  - Nodes: 855,802
  - Edges: 4,291,352
  - Description: Google web graph
- **WEB-WT10G**
  - Nodes: 1,458,316
  - Edges: 6,225,033
  - Description: TREC WT10G

**Bipartite affiliation (authors-to-papers) networks**
- **ATP-DBLP**
  - Nodes: 615,678
  - Edges: 944,456
  - Description: DBLP [21]
- **ATM-IMDB**
  - Nodes: 2,076,978
  - Edges: 5,847,693
  - Description: Actors-to-movies

**Internet networks**
- **ASSKITTER**
  - Nodes: 1,719,037
  - Edges: 12,814,089
  - Description: Autonomous systems
- **GNUTELLA**
  - Nodes: 62,561
  - Edges: 147,878
  - Description: P2P network [29]
Typical example: General Relativity collaborations (n=4,158, m=13,422)
More NCP Plots of Networks

(a) LIVEJOURNAL01
(b) MESSENGER-DE
(c) AT&T-DBLP
(d) CIT-HEP-TH
(e) WEB-GOOGLE
(f) AMAZONALL
NCP: LiveJournal ($n=5m$, $m=42m$)

- Better and better clusters
- Clusters get worse and worse
- Best cluster has ~100 nodes
As clusters grow the number of edges inside grows **slower** than the number crossing.

\[ \Phi = \frac{1}{7} = 0.14 \]

\[ \Phi = \frac{2}{10} = 0.2 \]

\[ \Phi = \frac{8}{20} = 0.4 \]

\[ \Phi = \frac{64}{92} = 0.69 \]

Each node has twice as many children.
Empirically we note that best clusters are barely connected to the network.
What If We Remove Good Clusters?

Nothing happens!  
⇒ Nestedness of the core-periphery structure
Denser and denser core of the network

Nested Core-Periphery (jellyfish, octopus)

Core contains 60% node and 80% edges

Whiskers are responsible for good communities

Whiskers
Communities: Issues and Questions
Communities: Issues and Questions

- **Some issues with community detection:**
  - Many different formalizations of clustering objective functions
  - Objectives are NP-hard to optimize exactly
  - Methods can find clusters that are systematically “biased”

- **Questions:**
  - How well do algorithms optimize objectives?
  - What clusters do different methods find?
Many Different Objective Functions

- **Single-criterion:**
  - Modularity: $m - E(m)$
  - Edges cut: $c$

- **Multi-criterion:**
  - Conductance: $c / (2m + c)$
  - Expansion: $c / n$
  - Density: $1 - m / n^2$
  - CutRatio: $c / n(N - n)$
  - Normalized Cut: $c / (2m + c) + c / 2(M - m) + c$
  - Flake-ODF: fraction of nodes with more than $\frac{1}{2}$ edges pointing outside $S$

$n$: nodes in $S$
$m$: edges in $S$
$c$: edges pointing outside $S$
Many algorithms to that implicitly or explicitly optimize objectives and extract communities:

- **Heuristics:**
  - Girvan-Newman, Modularity optimization: popular heuristics
  - Metis: multi-resolution heuristic [Karypis-Kumar ‘98]

- **Theoretical approximation algorithms:**
  - Spectral partitioning
NCP: Live Journal

\[ \varphi (\text{conductance}) \]

\[ \begin{align*}
10^0 & \\
10^{-1} & \\
10^{-2} & \\
10^{-3} & \\
10^{-4} & \\
\end{align*} \]

\[ n (\text{number of nodes in the cluster}) \]

\[ \begin{align*}
10^0 & \\
10^1 & \\
10^2 & \\
10^3 & \\
10^4 & \\
10^5 & \\
10^6 & \\
10^7 & \\
\end{align*} \]

Spectral

Metis
Properties of Clusters (1)

500 node communities from Spectral:

500 node communities from Metis:
Properties of Clusters (2)

- **Metis (red)** gives sets with better conductance
- **Spectral (blue)** gives tighter and more well-rounded sets
Multi-criterion Objectives

Observations:
- Conductance, Expansion, Norm-cut, Cut-ratio are similar
- Flake-ODF prefers larger clusters
- Density is bad
- Cut-ratio has high variance
Observations:
- All measures are monotonic
- **Modularity**
  - prefers large clusters
  - ignores small clusters